GUIDE PRATIQUE

LA RÉALISATION DES PROGRAMMES

MICHEL BENELFOUL



EDITIONS DU P.S.I.

GUIDE PRATIQUE

LA RÉALISATION DES PROGRAMMES

VOTRE BIBLIOTHEQUE D'INFORMATIQUE INDIVIDUELLE

Programmer en Assembleur — Alain Pinaud
Programmer en Basic — Michel Plouin
Le Basic et ses fichiers — Jacques Boisgontier
Programmer en L.S.E. — Stéphane Berche et Yves Noyelle
Programmer en Pascal — Daniel-Jean David et Jean-Luc Deschamps
Comment programmer Jean-Claude Barbance
Comprendre les microprocesseurs — Roland Dubois

La découverte de l'Applesoft - Frédéric Lévy et Dominique Schraen
La pratique de l'Apple II — Nicole Bréaud-Pouliquen

La pratique du LX500 — Alain Séméteys et Francis Vasse

La découverte du P.E.T./C.B.M. — Daniel-Jean David
La pratique du P.E.T./C.B.M. — Daniel-Jean David

La pratique du TRS-80 — Pierre Giraud et Alain Pinaud

ISBN : 2-86470-017-4

Imprimé en France

GUIDE PRATIQUE

LA RÉALISATION DES PROGRAMMES

MICHEL BENELFOUL

P.S.I.

EDITIONS DU P.S.I.

1981

Michel Benelfoul est informaticien. Il a connu chez Bull les tableaux de connexion avant d'aborder la programmation puis la conception d'applications de gestion.

Il est actuellement responsable de projets informatiques dans un établissement financier.

A mes compagnons des années folles,
les fans des "sixtys" ...

S O M M A I R E

AVANT PROPOS

La réalisation d'un programme exige de la méthode et de la précision. Les difficultés rencontrées sur de petits systèmes individuels sont les mêmes toutes proportions gardées bien entendu, que celles auxquelles sont confrontés les professionnels de l'informatique.

Il est donc intéressant pour l'amateur comme pour le futur informaticien d'acquérir d'emblée, une méthode d'analyse orthodoxe complétée en cela d'un support d'études et de programmation adapté.

Ce guide, que nous vous proposons, devrait répondre aux problèmes rencontrés dans l'utilisation d'un ordinateur individuel.

Cependant il n'a pas la prétention d'être une bible pour informaticien (il existe déjà de nombreux ouvrages sur le sujet et nous vous recommandons notamment aux éditions du P.S.I. "COMMENT PROGRAMMER" de Jean-Claude BARBANCE), mais plutôt d'être un auxiliaire simple, facile à consulter et peu volumineux, permettant de se familiariser rapidement à la pratique de l'analyse et de la programmation.

CONCEPTS DU LANGAGE

Dans toute technique en pleine évolution le vocabulaire n'est pas très précis et c'est là une difficulté majeure.

L'informatique n'échappe pas à la règle et il est bon de s'assurer, à tout moment, de la définition rigoureuse des termes utilisés.

Voici donc présenté, ci-après, un extrait de mots, de termes utilisés dans le contexte de l'informatique appliquée.

CONCEPTS DU LANGAGE - VOCABULAIRE

Informatique : Ensemble des techniques de la collecte, du tri, de la mise en mémoire, de la transmission et de l'utilisation des informations.

Elément : Partie constitutive d'un ensemble. Les données de l'information numériques et alphabétiques, les documents d'entrée et de sortie de l'ordinateur, les fichiers supports de l'information, les unités de traitement, la mémoire centrale, les unités logiques et arithmétiques, les périphériques.

Ensemble : Collection d'éléments, en nombre fini ou infini susceptibles de posséder certaines propriétés (dont le critère d'appartenance à cette collection est sans ambiguïté) et d'avoir entre-eux, ou avec des éléments d'autres ensembles, certaines relations (de N. BOURBAKI 1939).

Structure : Manière d'envisager un ensemble en fonction des lois de composition définies sur lui. Disposition des parties d'un ensemble abstrait, d'un phénomène ou d'un système complexe, généralement envisagée comme caractéristique de cet ensemble et comme durable.

Système : Ensemble coordonné de pratiques tendant à obtenir un résultat. Composé d'éléments ou constituants il est muni d'une structure lorsque : - ses constituants possèdent des propriétés en tant qu'appartenance au système ; - certaines de ces propriétés sont fonctionnelles, c'est-à-dire, affectent l'évolution du système. Il y a le système de traitement de l'information, le système d'exploitation.

Analyse : Décomposition d'un problème posé pour en déceler les éléments constituants et les liens qui les unissent en vue du traitement sur machine.
Un esprit est analytique s'il considère les choses dans leur élément (A. LALANDE).
L'esprit d'analyse doit se doubler d'un esprit de synthèse.
Ensemble, système formé de phénomènes solidaires, tel que "chacun dépend des autres et ne peut être ce qu'il est que dans et par sa relation avec eux". (A. LALANDE).
Dans la réalisation d'un système de traitement automatisé, l'analyse s'applique aux différentes étapes de la réalisation c'est ce que nous développons au chapitre traitant les concepts de l'analyse.

Logiciel : (Software en anglais). Ensemble de travaux de logique, d'analyse, de programmation, nécessaire au fonctionnement d'un système de traitement de l'information. Nous nous contenterons de l'expliquer en terme de programmation pour lequel il se compose de deux parties :

Le logiciel de base regroupant les programmes standardisés par le constructeur (d'un ordinateur donné) et mis à la disposition de l'utilisateur : programme de tri, d'interclassement, d'édition, de vidage de fichier sur imprimante, etc.

Le logiciel d'application regroupant les programmes de gestion conçus et écrits ou édités par l'utilisateur pour son usage propre.

Matériel : (Hardware en anglais). Désigne tout ce qui compose physiquement un ordinateur ; exemple : la mémoire à ferrites, à circuit imprimé, à disques, le pupitre de commande, les périphériques tels que dérouleurs de bande, unités de disques, câbles de liaison...

Avec les termes suivants, nous abordons la notion d'information d'une part, en désignant les supports qui servent de véhicules, d'autre part, les supports représentant l'information en sous ensembles logiques de traitement.

Les supports physiques d'entrée/sortie : Ce terme désigne le contenant et souvent aussi l'unité de lecture écriture correspondante

- clavier de terminal (support d'entrée uniquement)
- magnéto cassette
- unité de disque, disquette
- dérouleur de bande magnétique
- lecteur de carte perforée, de bande perforée (support d'entrée uniquement)

- perforateur de carte perforée (support de sortie uniquement
- écran de terminal
- imprimante

Les supports logiques d'entrée/sortie : Ce terme désigne le contenu. C'est le fichier de données, sa structure répond au découpage basé sur les niveaux de hiérarchie suivant :
- le *bloc*, de un à plusieurs enregistrements
- l'*enregistrement*, de un à plusieurs articles
- l'*article*, de une à plusieurs rubriques
- la *rubrique*, en Basic, elle est représentée par une variable.

C'est la plus petite unité d'information adressable par le programme.

L'algorithme : C'est un processus fini et déterminé. Il s'exprime par un nombre fixe de directives, dont la succession est rigoureuse. Le choix et l'ordre ne doivent, en aucun cas, être arbitraire.
Un programme peut contenir un algorithme de calcul, de contrôle, de formatage de données.

Le mode de travail : Le mode de *traitement par lot* dit *"temps différé"* par opposition au terme *"temps réel"* présenté plus loin; ce mode de travail correspond au traitement d'un lot de données d'une manière automatisée sans intervention manuelle entre le début et la fin du traitement.
Le mode de *traitement inter-actif* ou *conversationnel* dit *"temps réel"*, c'est un échange d'information instantané entre l'utilisateur et l'ordinateur.
Le dialogue s'effectue sous forme de questions-réponses.

Les fonctions automatisées désignent les opérations effectuées par l'ordinateur exclusivement ; exemple : lecture de

fichier, traitement des données (contrôle, calcul, formatage ...), impression de documents.

Les fonctions manuelles : Par opposition aux fonctions automatisées, c'est tout ce qui n'est pas traité par l'ordinateur.

La transaction : Messages échangés entre l'opérateur (à l'aide du clavier terminal) et l'ordinateur, à l'aide du programme contenant les questions et les réponses correspondantes à toutes les éventualités possibles.

La session : Traitement d'un certain nombre de transactions dans une unité de temps déterminé.

CONCEPTS DE L'ANALYSE

L'étude de conception et de réalisation d'un système de traitement de l'information, demande une approche méthodique des problèmes à étudier.

Nous pouvons distinguer 3 parties principales :

1- L'étude préalable
2- Le processus du choix
3- L'étude technique

L'ETUDE PREALABLE

Dites aussi analyse fonctionnelle.

C'est le regroupement de l'ensemble des tâches d'études administratives et techniques qui permettent de déboucher sur le choix d'une solution adaptée au problème posé.

Elle comprend les étapes suivantes :

L'ANALYSE QUALITATIVE

C'est l'étude des règles, des travaux, des documents et des structures.

L'ANALYSE QUANTITATIVE

C'est l'étude chiffrée des données, des résultats et des travaux.

LA REORGANISATION

Issue de l'étude de synthèse des deux premières étapes, elle peut déboucher sur la constitution de structures nouvelles, la création d'un système automatisé, ou la novation du système existant.

L'EVALUATION TECHNIQUE

Si le matériel existe déjà, elle consistera à vérifier qu'il a la capacité et les disponibilités de charge suffisantes pour traiter le problème et définir les éventuelles extensions.

Dans le cas contraire, trois directions sont à explorer :
- les caractéristiques du matériel et des conditions d'utilisation ;
- les problèmes de locaux inhérents au nouveau système et des incidences financières ;
- quelle sera la charge nouvelle en personnel ?

LE PROCESSUS DU CHOIX DU MATERIEL

A ce niveau, les études débouchent sur les contacts avec les fournisseurs.

C'est au cours de cette phase, que vont se concrétiser les engagements.

LE CAHIER DES CHARGES

Document destiné à la consultation des fournisseurs, il doit, en se fondant sur les procédures nouvellement créées, mettre en

évidence les caractéristiques souhaitées du matériel, ainsi que ses principes d'utilisation.

LE DEPOUILLEMENT DES OFFRES

C'est l'examen des propositions, étape délicate nécessitant une étude comparative des performances minitieuses à partir des critères de choix répondant aux intérêts de l'utilisateur.

LE CONTRAT D'ACQUISITION ET DE MAINTENANCE

Un document doit être produit définissant les relations entre le constructeur et l'utilisateur.

ETUDE TECHNIQUE

Dite aussi analyse organique.

Un système informatique doit donner à ses utilisateurs toutes les garanties de fiabilité de l'information à traiter.

D'abord et surtout les informations d'entrée du système. Pour cela elles doivent être classifiées selon des critères d'appréciation ne laissant aucune ambiguïté sur leur forme et leur contenu.

En premier lieu, l'information doit être répertoriée dans l'une des catégories suivantes :

1- **Information qualitative** : elle correspond à l'identification d'un objet ou d'un individu. C'est généralement le nom, l'adresse, la profession, un code se rapportant au caractère de l'objet ou de l'individu.

2- **Information quantitative** : elle correspond à la notion de nombre, un montant, une quantité.

Son contenu doit être défini comme suit :
- valeur alphabétique : le nom d'un individu ;
- valeur alpha-numérique : le numéro de compte bancaire ;
- valeur numérique : un montant.

LES CONTROLES

La définition des informations dans leur caractère comme dans leur contenu précise le type de contrôle à effectuer sur chaque variable avant de les considérer fiables pour la suite du traitement.

Le contrôle de validité

Il consiste à s'assurer qu'une variable possède un contenu rigoureusement conforme à l'un de ceux qui est attendu.

Exemple : Dans le programme de facturation donné au chapitre 4, le code client peut prendre l'une des valeurs suivantes : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, correspondantes aux 10 clients possibles. Tout autre valeur pourrait être refusée par un contrôle adéquat avec visualisation d'un message d'erreur et retour sur la séquence d'introduction du code client.

Le contrôle de vraisemblance

Il consiste à s'assurer que le contenu d'une variable est compris dans des limites données.

Exemple : l'introduction d'une date comprend le jour, le mois et l'année. Pour le jour, la valeur introduite doit être comprise entre 01 et 31.

Le contrôle de compatibilité

Il consiste à s'assurer que plusieurs variables possèdent un contenu conforme à une relation attendue entre elles.

Exemple : Dans le cas de notre programme de facturation, on aurait pu insérer un contrôle de compatibilité entre la variable Q1 et Q2 ; Q1 représentant un nombre de paires de skis, Q2 un nombre de fixations et, en considérant que Q1 ne doit jamais être supérieur à Q2.

LES FICHIERS

L'information est stockée sur des supports logiques appelés fichiers.

Ces fichiers ont selon leur rôle une définition particulière.

Les fichiers maîtres ou fichiers dits "permanents"

Ils contiennent les informations essentielles à l'identification d'un individu. Ce type de fichier sert de référence lors des traitements de contrôle, de mise à jour et de consultation.

La rétention de ce type de fichier est liée à sa durée d'utilisation, il faut prévoir par sécurité la copie de tel fichier.

Les fichiers "mouvements"

Ils contiennent les informations se rapportant à la création, la modification et l'annulation d'individus.

Ils sont rapprochés du fichier maître à l'aide des unités de traitement de contrôle et de mise à jour. En principe la rétention de ce type de fichier est liée à la périodicité même des traitements.

Autrement dit un fichier "mouvements" en chasse un autre.

Les fichiers "intermédiaires" dits aussi fichiers locaux

Ils servent de lien entre deux unités de traitement. Par exemple entre deux éditions dont les critères de classement sont différents, une unité de tri intercalée créera un fichier intermédiaire.

Sa rétention est très courte, en général le temps de son utilisation dans l'unité de traitement suivante.

Les fichiers "répertoire"

Ils servent généralement pour des consultations ponctuelles, et ne contiennent que des informations de référence complémentaires, et non prévues sur les fichiers maîtres.

LES UNITES DE TRAITEMENT

Chaque unité de traitement est un ensemble de fonctions assemblées les unes à la suite des autres, le tout représente un programme.

Définir une unité de traitement, c'est délimiter avec précision les fonctions de saisie, de contrôle, de calcul, d'impression de mise à jour, définir la forme et le contenu des informations introduites et produites depuis les supports logiques d'entrée et de sortie et les documents imprimés.

Un système de traitement de l'information, peut se composer d'un nombre d'unités représentatives chacune d'une fonction spécifique au système.

Exemples

- U.T de saisie des données en mode interactif
- U.T de contrôle des données
- U.T. de calcul
- U.T. de mise à jour du fichier maître
- U.T. d'édition des mouvements validés.

SYMBOLES FONCTIONNELS

Pour représenter fonctionnellement les unités de traitement, ainsi que leur enchaînement, des symboles normalisés ont été définis. Les symboles sont les suivants :

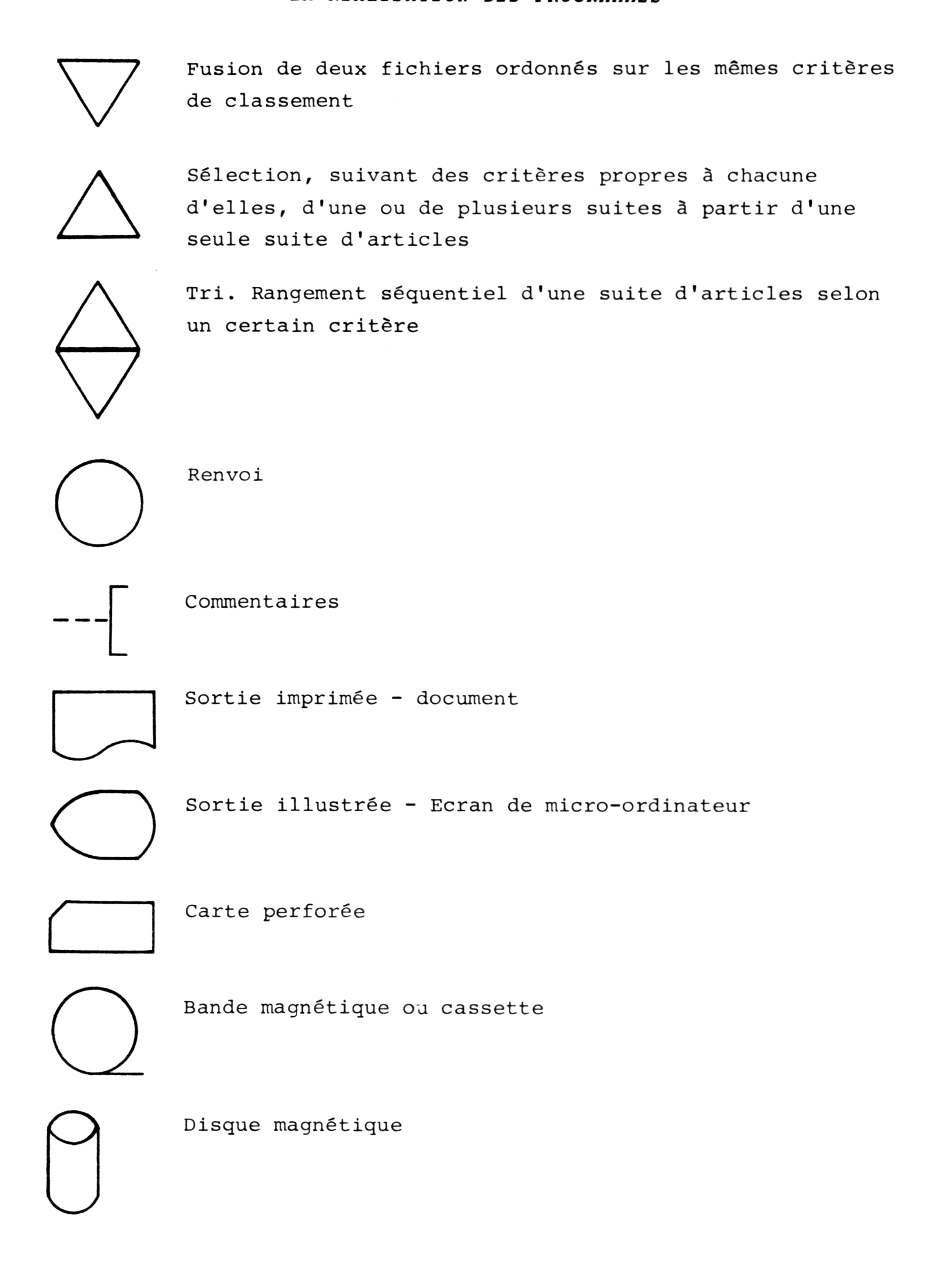
Opération manuelle

Entrée manuelle au moyen d'un clavier

Ligne de liaison. Transfert d'information

Transfert d'information par une télécommunication

Opération effectuée avec l'ordinateur



Fusion de deux fichiers ordonnés sur les mêmes critères de classement

Sélection, suivant des critères propres à chacune d'elles, d'une ou de plusieurs suites à partir d'une seule suite d'articles

Tri. Rangement séquentiel d'une suite d'articles selon un certain critère

Renvoi

Commentaires

Sortie imprimée - document

Sortie illustrée - Ecran de micro-ordinateur

Carte perforée

Bande magnétique ou cassette

Disque magnétique

Pour illustrer le choix des symboles d'un organigramme, selon les références de l'AFNOR, nous avons représenté le système de gestion d'une entreprise prenant en charge les commandes, la facturation et la livraison de matériel de skis, auprès d'un certain nombre de points de vente.

Le système de gestion comprend :

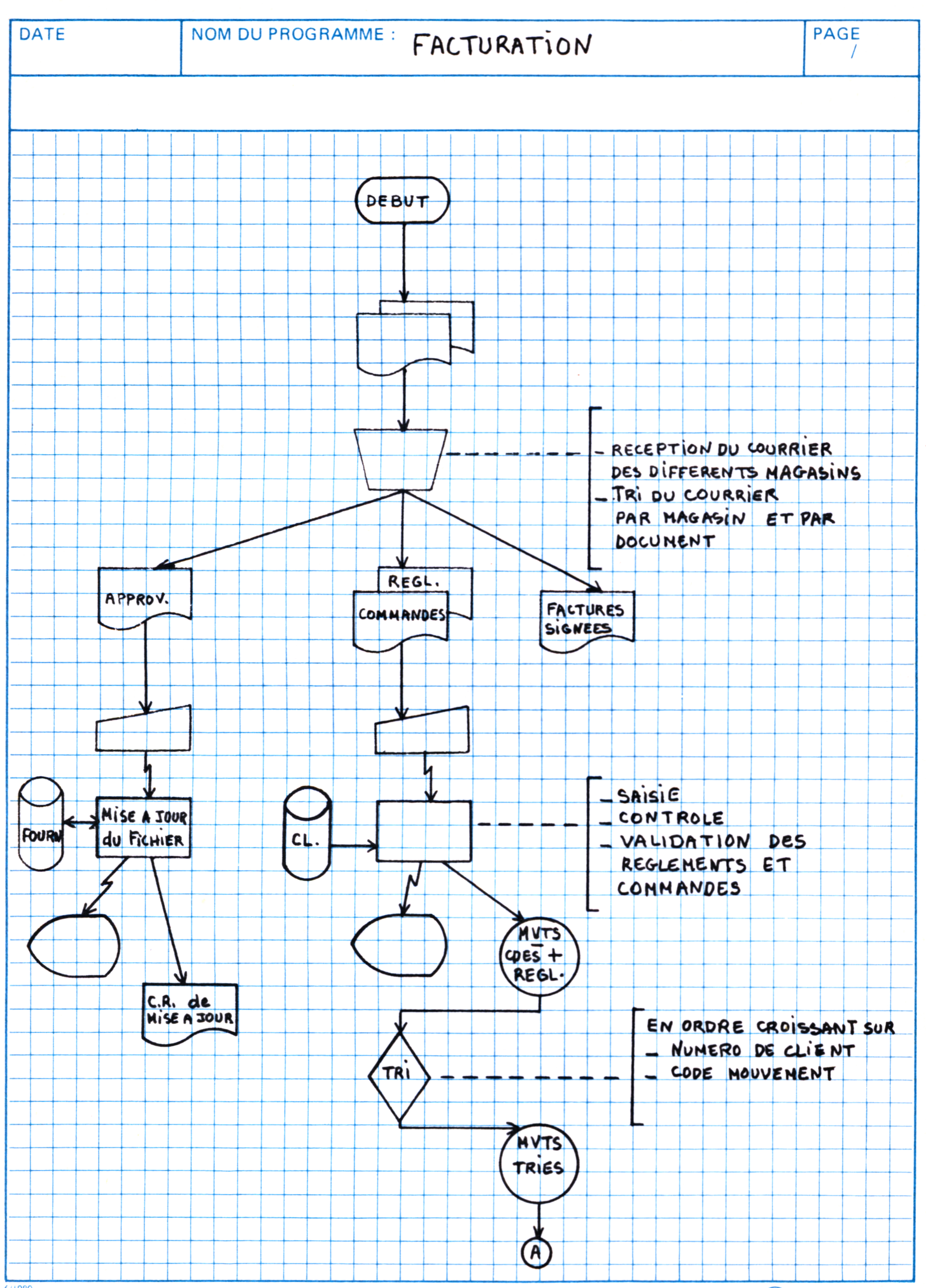
- un fichier des fournitures
- un fichier clients
- un fichier factures

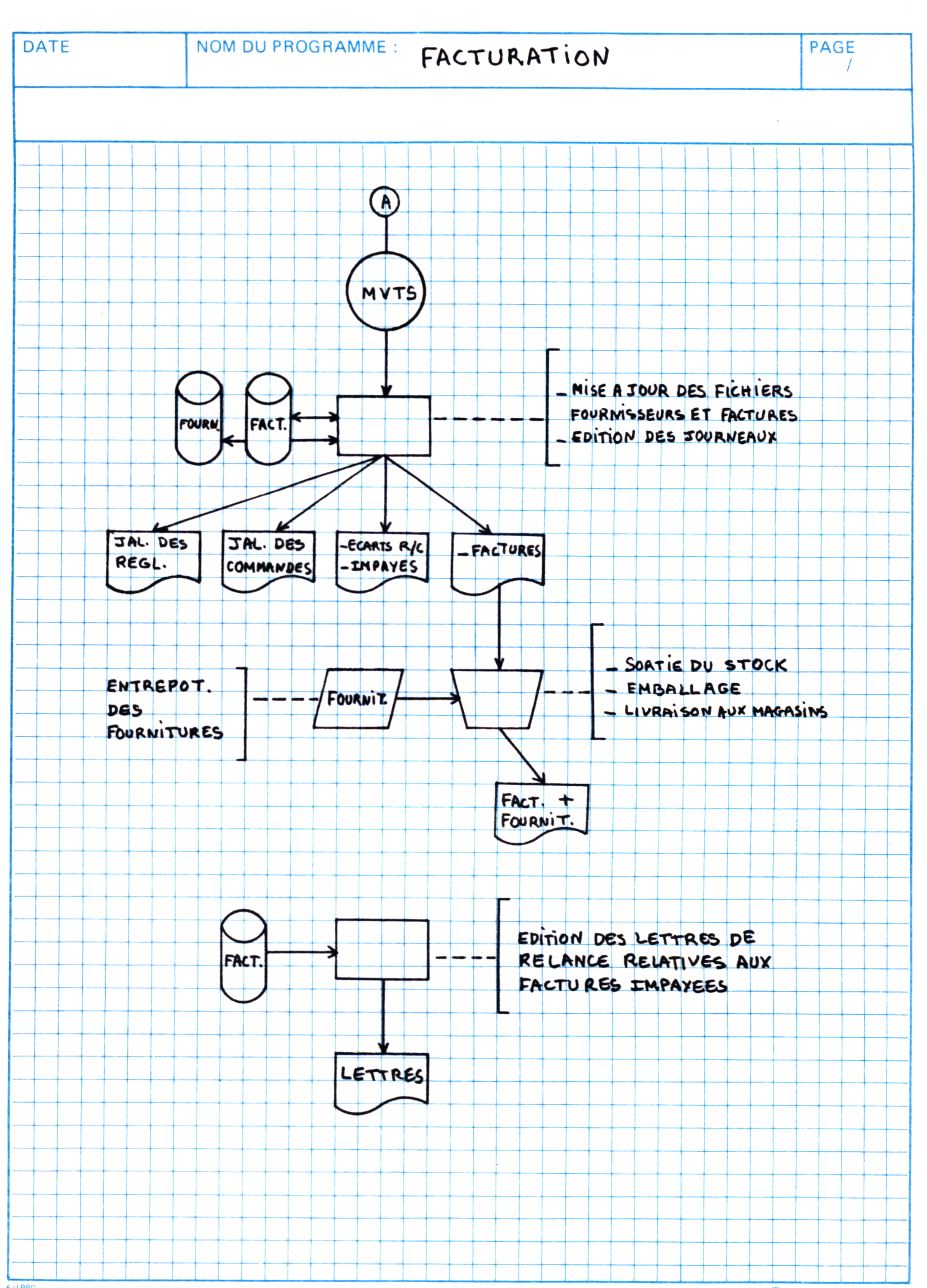
Les mouvements d'entrée sont :

- les mouvements de réapprovisionnement du fichier fournitures
- les bons de commande
- les règlements
- les factures signées par les clients

Les mouvements de sortie :

- les factures
- le journal des règlements
- le journal des commandes
- les relances de factures impayées et les écarts de règlement
- les états de mise à jour des fichiers.


DATE
NOM DU PROGRAMME :
FACTURATION
PAGE
1
DEBUT
- RECEPTION DU COURRIER
DES DIFFERENTS MAGASINS
- TRI DU COURRIER
PAR MAGASIN ET PAR
DOCUMENT
APPROV.
REGL.
COMMANDES
FACTURES
SIGNEES
FOURN
MISE A JOUR
du FICHIER
CL.
- SAISIE
- CONTROLE
- VALIDATION DES
REGLEMENTS ET
COMMANDES
MVTS
CDES +
REGL.
C.R. de
MISE A JOUR
TRI
EN ORDRE CROISSANT SUR
- NUMERO DE CLIENT
- CODE MOUVEMENT
MVTS
TRIES
A
6/1980


DATE
NOM DU PROGRAMME : FACTURATION
PAGE
/
A
MVTS
FOURN.
FACT.
_ MISE A JOUR DES FICHIERS FOURNISSEURS ET FACTURES
_ EDITION DES JOURNEAUX
JAL. DES REGL.
JAL. DES COMMANDES
- ECARTS R/C
- IMPAYES
_ FACTURES
ENTREPOT. DES FOURNITURES
FOURNIT.
_ SORTIE DU STOCK
_ EMBALLAGE
_ LIVRAISON AUX MAGASINS
FACT. + FOURNIT.
FACT.
EDITION DES LETTRES DE RELANCE RELATIVES AUX FACTURES IMPAYEES
LETTRES
6/1980
Reproduction interdite (c) Editions du P.S.I.

LES TABLES DE DECISION

La réalisation d'un programme demande le respect de certaines règles quasiment immuables d'un programme à l'autre.

Un développement précis est présenté dans les pages suivantes, mais en avant propos, quelques recommandations ne sont pas inutiles.

Définir l'environnement du programme, c'est-à-dire, bien connaître le matériel prévu pour le supporter.

Se poser en préalable la question suivante : "Est-ce que le problème demandé est compatible dans sa réalisation avec le matériel choisi ?"

Le profil de la solution doit se définir en commençant par la fin, autrement dit, les résultats à obtenir.

Quand l'inventaire exhaustif des résultats est effectué, on dresse un tableau d'équivalence avec les données disponibles à l'entrée du programme. A ce niveau l'utilisation des tables de décision est bien utile.

Le cheminement est parfois complexe entre les données d'entrée (le produit brut) et les résultats de sortie (le produit fini).

Une table de décision est la représentation schématique sous forme de tableau d'une série particulière de :

- conditions (les données)
- actions (les opérations à exécuter)
- règles (opérations dépendant de certaines conditions)

	Règles
Enoncé des conditions	Réalisation des conditions
Enoncé des actions	Réalisation des actions

Il y a trois types de table

- les tables limitées
- les tables étendues
- les tables mixtes

LES TABLES LIMITEES

L'énoncé des conditions définit à la fois une variable et la valeur de cette variable, et l'énoncé des actions définit entièrement les actions. La réalisation des conditions s'exprime simplement par les mots OUI ou NON.

Exemple

	R0	R1	R2	R3	R4	R5	R6	R7
Condition a	Oui	Oui	Oui	Non	Non	Non	Non	Oui
Condition b	Oui	Oui	Non	Non	Oui	Oui	Non	Non
Condition c	Oui	Non	Non	Non	Non	Oui	Oui	Oui
Action 1	X							X
Action 2		X				X		
Action 3			X				X	
Action 4					X	X		

Règle 0 = Les conditions a, b, c, sont réalisées, il faut effectuer l'action 1.

Règle 1 = Les conditions a, b, sont réalisées, mais pas la condition c, il faut effectuer l'action 2.

Règle 2 = La condition a est seule réalisée, il faut effectuer l'action 3.

Règle 3 = Aucune condition n'est réalisée, aucune action ne doit être effectuée.

Règle 4 = La condition b est seule réalisée, il faut effectuer l'action 4.

Règle 5 = Les conditions b et c sont réalisées mais pas la condition a, il faut effectuer les actions 2 et 4.

Règle 6 = La condition c est seule réalisée, il faut effectuer l'action 3.

Règle 7 = Les conditions a et c sont réalisées mais pas la condition b, il faut effectuer l'action 1.

Remarque : Pour n conditions, il est possible d'avoir 2^n régles.

LES TABLES ETENDUES

L'énoncé des conditions ne définit pas la valeur d'une variable et l'énoncé des actions ne précise pas complètement l'action à exécuter. Aussi est-il nécessaire d'apporter ces compléments d'information pour chacune des règles.

Exemple :

	R0	R1	R2
Condition a	$0 < x \leq 10$	$10 < x \leq 50$	$50 < x < 100$
Condition b	y=1	y=2	y=0
Action 1	S1	S2	S2
Action 2	S3	S4	S5

Règle 0 = Si x est supérieur à 0 et inférieur ou égal à 10, si y est égal à 1, il faut exécuter les séquences S1 et S3.

Règle 1 = Si x est supérieur à 10 et inférieur ou égal à 50, si y est égal à 2, il faut exécuter les séquences S2 et S4.

Règle 2 = Si x ext supérieur à 50 et inférieur ou égal à 100, si y est égal à 0, il faut exécuter les séquences S2 et S5.

LA TABLE MIXTE

La table mixte comporte des parties "limitées" et des parties "étendues".

Exemple :

	R0	R1	R2
Condition A	$0 < x \leq 10$	$10 < x \leq 50$	$50 < x \leq 100$
Condition B	Oui	Oui	Non
Action 1	S1	S1	S4
Action 2	S2		S6

Règle 0 = Si x est supérieur à 0 et inférieur ou égal à 10, si la condition B est réalisée il faut exécuter les séquences S1 et S2.

Règle 1 = Si x est supérieur à 10 et inférieur ou égal à 50, si la condition B est réalisée il faut exécuter la séquence S1.

Règle 2 = Si x est supérieur à 50 et inférieur à 100 si la condition B n'est pas réalisée, il faut exécuter les séquences S4 et S6.

POURQUOI DES TABLES DE DECISION ?

Elles synthétisent les éléments d'appréciation (énoncé des conditions, les règles, énoncé des actions) sous une forme rigoureuse et compatible avec le langage de programmation.

QUAND FAUT-IL LES UTILISER ?

Quand la réalisation d'une action ou d'une combinaison d'actions exigent l'analyse d'un ensemble de conditions complexes avant leur représentation par l'organigramme. Elles sont également très utiles lors de la mise au point du programme, en permettant de vérifier que tous les cas possibles sont bien contrôlés. (voir page 76)

METHODE DE REALISATION D'UN PROGRAMME

La réalisation d'un programme passe par un certain nombre d'étapes à exécuter dans un ordre chronologique donné.

EXPOSE DU PROBLEME

- de quoi s'agit-il ?
- comment le traiter ?

A la première question, on doit répondre par la définition précise des besoins pouvant comporter les paragraphes suivants :

- la définition du sujet à traiter ;
- le volume des transactions ;
- la périodicité des traitements ;
- le délai de remise des résultats ;
- avantages et inconvénients de la solution.

La deuxième question exige un ensemble de réponses. Chaque réponse doit faire l'objet d'un développement organique avec la mise en évidence des aspects techniques propres au matériel choisi.

INVENTAIRE DES RESULTATS A OBTENIR
(données de sortie)

Sous quelle forme ?

- Dessiner sur grille d'impression les résultats.
- Définir les variables correspondant aux sorties sur écran et/ou imprimante.
- Définir les fichiers résultats sur cassette et/ou disquette.
- Définir les variables propres aux fichiers.

INVENTAIRE DES DONNEES D'ENTREE

Depuis le clavier

- Dessiner les masques de saisie sur grille d'écran.
- Définir les variables correspondantes.
- Définir les contrôles de saisie.

Depuis les fichiers (cassette et/ou disquette)

avec les données d'entrée,
avec les données des autres fichiers.

- Décrire leur contenu.
- Définir les variables correspondantes.
- Définir les relations logiques des données.

VOLUME DES DONNEES A TRAITER

- Calcul du nombre total de caractères à saisir.
- Calcul du nombre total de caractères des fichiers d'entrée.
- Calcul du nombre total de caractères des fichiers de sortie, sur cassette, sur disquette, sur imprimante.

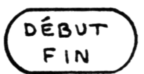
LE DECOUPAGE DU TRAITEMENT EN BLOCS DE FONCTION

Dessiner l'organigramme

- Début (Mise à l'Etat Initial des variables concernées).

- Saisie au clavier (INPUT, GET..).
- Lecture de fichiers.
- Traitement des données (contrôle, calcul, comparaison, transfert, R.E.I. de variables ...).
- Edition (sur écran, sur imprimante).
- Ecriture de fichiers.
- Fin de programme.

SYMBOLES POUR ORGANIGRAMME DE PROGRAMMATION



Début ou fin d'un organigramme



Entrée - Mise à disposition d'une information à traiter.
Sortie - Enregistrement d'une information traitée.



Opération qui détermine partiellement ou complètement la voie à suivre dans un embranchement ou un sous-programme. Il est également utilisé pour mettre un aiguillage en position.



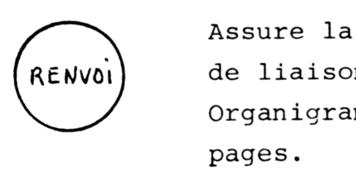
Exploitation de conditions multiples impliquant le choix d'une voie parmi plusieurs.

TRAITEMENT

Opération ou groupe d'opérations portant sur des données à traiter pour lesquelles un symbole particulier n'est pas défini.

SOUS-PROGRAMME

Ensemble d'instructions représentant une ou plusieurs fonctions accessibles depuis l'un des niveaux quelconques du programme.



Assure la continuité du traitement lorsqu'une ligne de liaison ne peut être représentée par exemple. Organigramme de programmation exprimé sur plusieurs pages.

Commentaires.

DEBUT DE PROGRAMME

DEBUT

Un programme bien structuré se découpe comme un livre en chapitres, paragraphes, voir alinéas.

Le chapitre début doit contenir 2 paragraphes.

P1/L'identification du programme - Préambule

P11 - le titre

P12 - le nom de l'auteur

P13 - la date de réalisation

P14 - le numéro de version

P15 - la configuration du matériel nécessaire à son fonctionnement.

P2/La mise à l'état initial des variables - introduction

Cette partie est plus délicate car elle dépend des traitements réalisés dans les différents blocs de fonction exécutés aux chapitres suivants.

Par exemple l'utilisation cohérente d'un indice ne peut s'effectuer que si il a une valeur de départ égale à 10. La tâche de ce paragraphe est donc de lui attribuer d'office cette valeur avant sa toute première utilisation.

Une variable est utilisée comme constante ; elle contient un taux représentant la T.V.A. sur un produit donné. La fonction M.E.I. va lui attribuer au départ une valeur qui restera immuable pendant tout le déroulement du programme.

Les variables utilisées pour effectuer des cumuls nécessitent avant toute opération arithmétique leur forcage à zéro sous peine d'avoir des incidents de logiciel[*], ou bien même des erreurs de cumul si ces variables contiennent des résultats de traitements précédents.

** C'est vrai pour certains ordinateurs. Le cumul de valeurs numériques dans une zone "à blanc", provoque de graves incidents.*

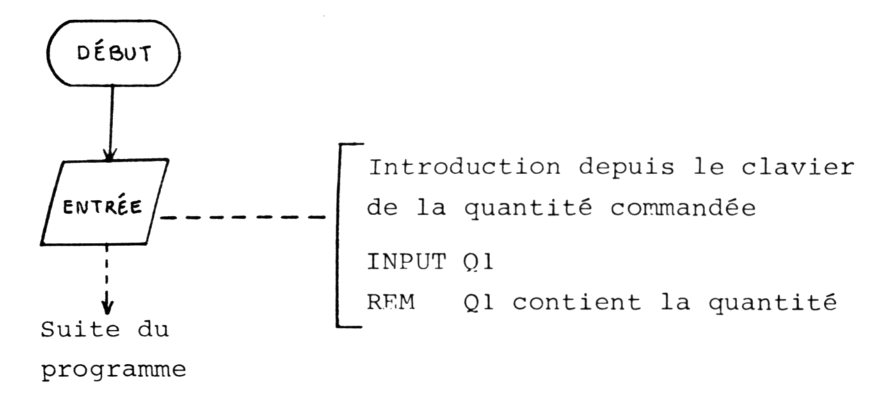
Le cumul arithmétique à plusieurs niveaux hiérarchiques demande une M.E.I. des variables concernées, mais aussi une remise à l'état initial (R.E.I.) conditionnée, et relative aux variables propres à chaque niveau hiérarchique.

La fonction R.E.I. s'effectue dans les blocs de traitement vus plus loin.

ENTREE
SORTIE

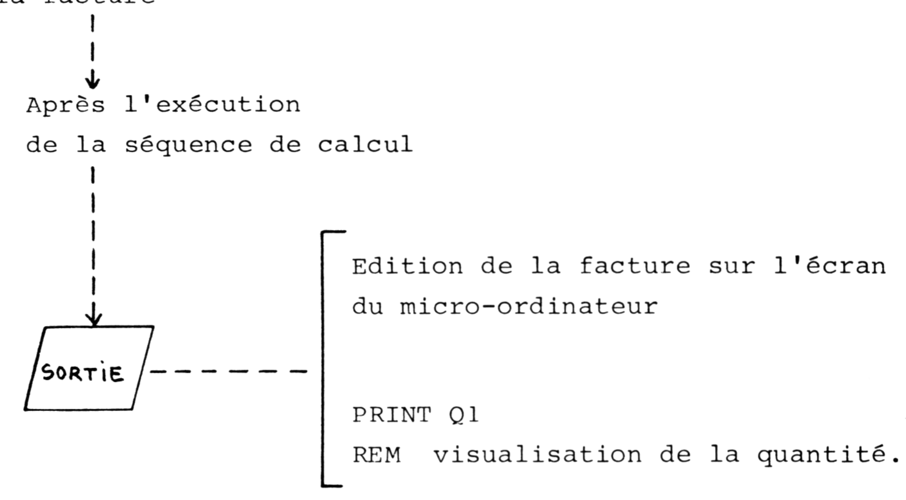
Ce symbole représente pour la *partie entrée* la fonction de saisie d'une information, ou un ensemble d'informations depuis le clavier, ou depuis un fichier magnétique sur disquette ou sur cassette.

Exemple : Dans un programme de facturation, il est demandé d'introduire la quantité commandée



Pour la *partie sortie,* le symbole représente un écran, une imprimante, ou un fichier magnétique disquette ou cassette.

Exemple : Après la séquence de calcul, on procède à l'édition de la facture



LA PREPARATION

A:: B

Le symbole préparation se traduit en langage de programmation par une instruction - IF - introduisant la notion de comparaison avant l'exécution conditionnelle du traitement.

L'expression symbolique A :: B se lit "comparer A à B". Les termes à comparer doivent avoir les mêmes caractéristiques ; la cohérence du résultat en dépend. On ne compare pas des litres et des kilogs.

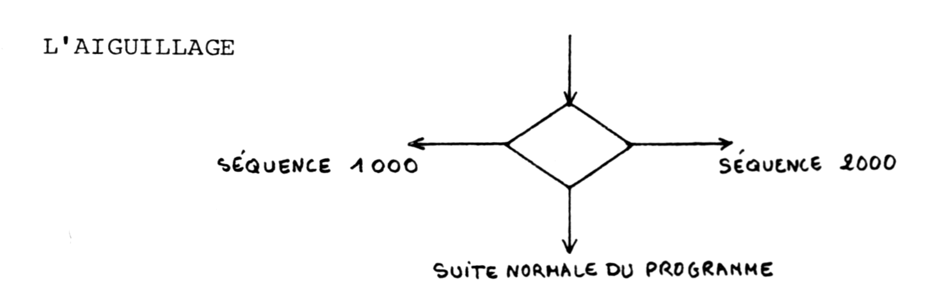
Il est donc très important que le contenu des variables à comparer soit homogène ; elles peuvent être classées sous 3 types de valeur possibles :

- valeur purement numérique (Ø à 9 exclusivement) ;
- valeur entière (Ø à 9 sans chiffre après la virgule - Basic) ;
- valeur algébrique (Ø à 9 avec le signe + ou -) ; le Basic considère souvent implicitement les valeurs numériques signées ;
- valeur alpha-numérique (combinaison de valeurs alphabétiques, valeurs numériques, caractères spéciaux, espace, point, etc.).

Il est nécessaire de prévoir en séquence début la M.E.I. des variables utilisées dans les termes à comparer, car les programmeurs tombent souvent dans le chausse-trappe suivant :

- lors de la comparaison de 2 termes considérés purement numériques, si l'un des 2 n'a pas été initialisé, c'est-à-dire garni de zéros, et qu'il n'a pas fait l'objet au cours du traitement d'un chargement de valeurs numériques, il ne contient que la valeur - blanc - et bien souvent le résultat de la comparaison renvoie sur un branchement conditionnel inverse de celui souhaité.

L'AIGUILLAGE



Comme son nom l'indique il est à la croisée des chemins. Comme le voyageur arrivé à un carrefour, le programme doit y prendre une décision sur la direction à suivre : à gauche, à droite, ou bien tout droit, c'est-à-dire, ici, aller à l'adresse 1000, ou l'adresse 2000 ou bien continuer tout simplement le déroulement des instructions suivantes ; équivalent à aller tout droit pour le voyageur.

Pour illustrer le rôle joué par l'aiguillage, voici un exemple de fonctionnement simulé d'une gare de triage.

Exposé du problème

Un poste de triage doit sélectionner des wagons en fonction de leur destination et leur caractéristique pour les "aiguiller" sur leur voie d'affectation.

Les critères de sélection sont :

1- le code direction - variable DØ.

2- le code caractère du wagon - marchandises - , - voyageur - variable CØ.

Le tableau de formation des trains s'établit comme suit :

Direction	Code direction	Code caractère	Voie d'affectation
ANGOULEME	ANG	M	1
NANTES	NAN	M	2
LILLE	LIL	V	3
BORDEAUX	BOR	V	4
NANTES	NAN	V	5
NANCY	NCY	V	6
BORDEAUX	BOR	M	7
Sans affectation	-	-	8

DATE

NOM DU PROGRAMME : TRIAGE DES WAGONS

PAGE 1

ORGANIGRAMME

ARRIVÉE DES WAGONS A TRIER

Aiguillage -D$-

ANG — OUI → VOIE 1 ; NON

LIL — OUI → VOIE 3 ; NON — -D$-

NCY — OUI → VOIE 6 ; NON — -D$-

NAN — OUI ; NON — -D$-

Aiguillage -C$

M — OUI → VOIE 2 (WAGONS DE MARCHANDISES) ; NON

V — OUI → VOIE 5 (WAGONS DE VOYAGEURS) ; NON → VOIE 4 — -C$-

BOR — OUI ; NON → VOIE 8 — D$

V — OUI → VOIE 4 ; NON — -C$-

M — OUI → VOIE 7 ; NON → VOIE 8 — -C$-

VOIE 1 | VOIE 3 | VOIE 6 | VOIE 2 | VOIE 5 | VOIE 4 | VOIE 7 | VOIE 8

Pour certains wagons leur affectation ne nécessite le passage que sur un seul aiguillage, alors que pour d'autres 2 aiguillages sont nécessaires, notamment les wagons à destination de Nantes et Bordeaux.

La programmation de l'organigramme (page suivante) devient à ce niveau une formalité et une garantie de bon fonctionnement du premier coup !!!

DATE

NOM DU PROGRAMME
TRIAGE DES WAGONS

PAGE
1

FEUILLE DE PROGRAMMATION BASIC

Numéro de ligne — NUMEROS DES COLONNES ECRAN — LIGNE ECRAN

```
100 IF D$="ANG"THEN PRINT"VOIE 1"
110 IF D$="LIL"THEN PRINT"VOIE 3"
120 IF D$="NCY"THEN PRINT"VOIE 6"
130 IF D$="NAN"ANDC$="M"THENPRINT"VOIE 2"
140 IF D$="NAN"ANDC$="V"THENPRINT"VOIE 5"
150 IF D$="BOR"ANDC$="V"THENPRINT"VOIE 4"
160 IF D$="BOR"ANDC$="M"THENPRINT"VOIE 7"
170 REM *** L'INSTRUCTION SUIVANTE INDIQUE
180 REM *** LES WAGONS SANS AFFECTATION.
190 PRINT"AU DEPOT VOIE 8"
```

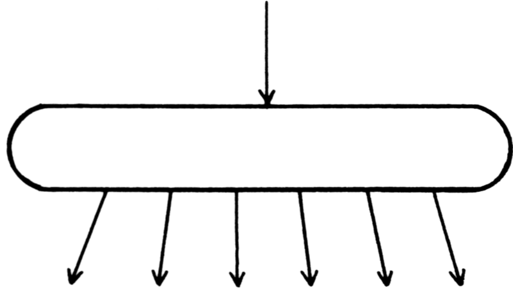
Certains Basic plus performant possèdent pour les branchements conditionnels une fonction supplémentaire (il faudrait dire plutôt appréciation), il s'agit de la condition - ELSE - (sinon).

Autrement dit :

IF(Si..) D$ = "NAN" AND C$ = "M" THEN (Alors je fais..) PRINT"VOIE2" ELSE (SINON je fais ...) PRINT"VOIE5"

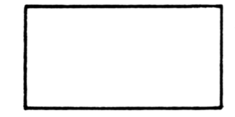
Cette subtilité permet donc la concaténation de 2 instructions - IF - en une seule et d'aller dans 2 directions possibles avant d'aller en séquence (c'est-à-dire tout droit). Dans ce cas les trois sorties du losange peuvent être utilisées.

Il existe également, en Basic, une possibilité de test unique avec possibilité de renvois multiples. C'est l'instruction ON ... GOTO ... ou ON ... GOSUB ... Dans ce cas le test de la valeur d'une variable numérique pourra renvoyer dans un nombre de directions supérieur à trois. On utilisera alors le symbole suivant :



Ces instructions très puissantes nécessitent, en général, une "préparation" importante. Par exemple, dans le cas des trains on aurait pu calculer le numéro de voie et faire un seul test de triage.

TRAITEMENT



Un traitement reflète une instruction ;

Exemple : Calculer la T.V.A. sur un total de produits commandés

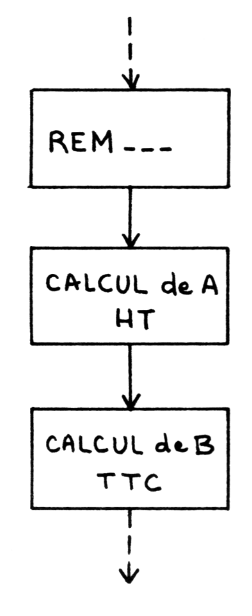
```
100  A = Q1 * 17.60
```

ou bien un ensemble d'instructions

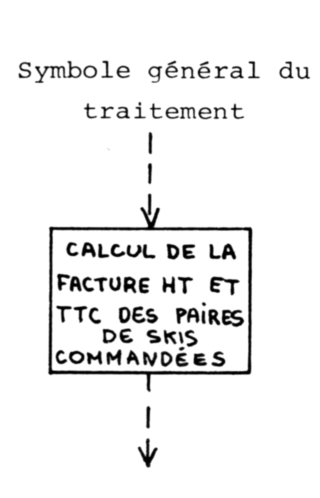
Exemple :

```
100  REM Q1 Représente le nombre de paires de skis
110  REM P2 Représente le prix unitaire d'une paire de
            skis
120  REM T1 Représente la T.V.A.
130  REM A  Représente le prix de la facture H.T.
140  REM B  Représente le prix de la facture TTC
150  A = Q1 * P1
160  B = A * T1
```

Symboles détaillés du traitement



Symbole général du traitement



LE SOUS-PROGRAMME

Le sous-programme représente une instruction ou un ensemble d'instructions implanté à un niveau quelconque du programme, auquel on peut accéder depuis n'importe quel autre niveau.

Il évite la répétition intempestive d'une fonction dont l'utilisation intéresse et répond aux besoins particuliers des traitements situés à des niveaux différents dans le programme.

En Basic l'accès au sous-programme s'effectue avec l'instruction GOSUB xxxx. Littéralement signifie aller à la subroutine xxxx. Ce qui en français se traduit par sous-programme.

Le retour à la séquence de traitement d'origine est assumé par l'instruction RETURN qui est la dernière instruction impérative dans un sous-programme.

Dans quelle circonstance le programmeur est-il amené à considérer qu'un traitement donné peut être représenté dans un sous-programme ?

Prenons l'exemple du calcul et de l'édition d'un facture sur l'écran d'un P.S.I. qui ne possède pas un logiciel suffisamment puissant pour assumer le cadrage automatique des montants à éditer.

La facture comprend plusieurs lignes, à raison d'une ligne par produit différent. Chaque produit fait l'objet d'un calcul et le résultat doit être édité avec une virgule et deux décimales ; mettre un zéro à gauche de la décimale si le résultat ne comporte pas d'entiers. (1)

---

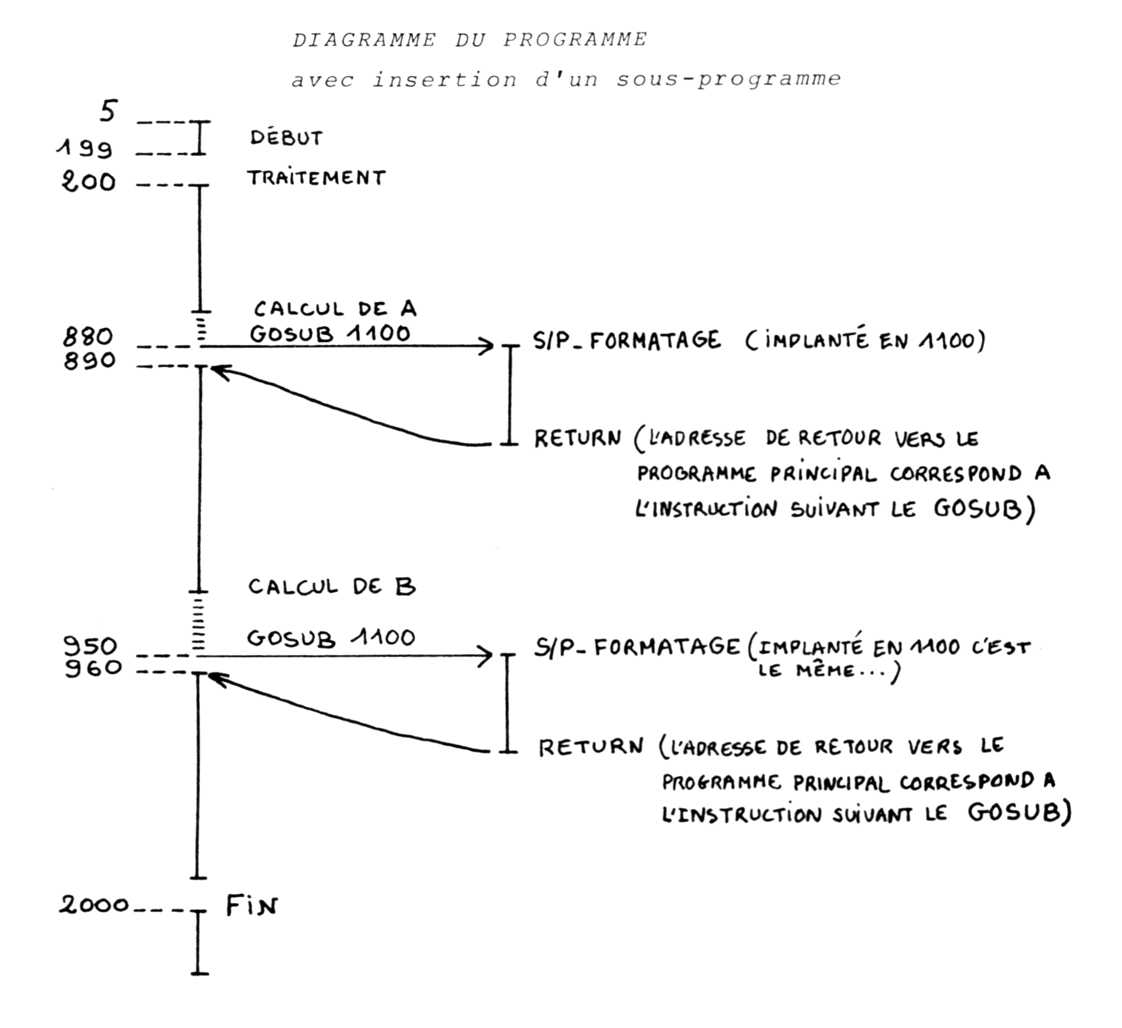
(1) Rassurez-vous, la solution programmée est donnée au chapitre 8 traitant de l'exemple de réalisation d'un programme.

A l'issue de chaque fonction de calcul relative à chaque produit, il est bien évident que le traitement de formatage de la variable contenant le résultat est extrêmement complexe et qu'il serait absurde de répéter d'une manière linéaire cette séquence autant de fois qu'il y a de lignes à éditer.

C'est là qu'intervient la notion de sous-programme qui va permettre au programmeur de gagner de la place en mémoire et du temps de programmation.

Le diagramme suivant illustre les relations entre les séquences dites linéaires et le sous-programme.

DIAGRAMME DU PROGRAMME
avec insertion d'un sous-programme



LES SAUTS DE SEQUENCE



Ce symbole est utilisé chaque fois que la liaison directe par un trait n'est pas possible, ou pas pratique, dans le cas des branchements :

1- GOTO "ALLER A" une adresse du programme principal en sautant une ou plusieurs instructions soit en amont soit en aval du saut.

2- GOSUB "ALLER A" une adresse correspondant à la première instruction d'un sous-programme.

3- RETURN "RETOUR A" dernière instruction d'un sous-programme renvoi dans le programme principal à l'instruction suivant immédiatement le GOSUB d'origine.

1

A : : B

Oui

2

<

Renvoi par THEN GOTO au niveau du traitement 2

Non

Oui

3

>

Renvoi par THEN GOTO au niveau du traitement 3

Non

TRAITEMENT 1

1

Renvoi par GOTO au niveau Préparation

2

TRAITEMENT 2

RENVOI

SOUS-PROGRAMME

3

TRAITEMENT 3

- Renvoi par GOSUB au S/Programme
- Puis retour par RETURN au trait-2

FIN DE PROGRAMME

FIN

Avec ce dernier symbole, nous sommes à l'épilogue de l'histoire, disons, plus prosaïquement, la fin du programme.

Que peut contenir ce chapitre ? Par exemple, une séquence d'édition indiquant justement que le programme est terminé.

Si c'est un programme de jeu, on peut imaginer le scénario suivant :

PRINT "LA PARTIE EST TERMINEE VOULEZ-VOUS RECOMMENCER?" si la réponse est négative une séquence d'impression confirmera le souhait du joueur.

PRINT "LE PROGRAMME EST TERMINE PAR L'INSTRUCTION STOP"

En fait, l'instruction Basic de fin de programme est - END -

Le microprocesseur, lorsqu'il rencontre cette instruction interromp le déroulement du programme et affiche le message - READY -

Une autre fonction propre au traitement de fin de programme consiste à éditer un bilan de traitement qui n'a de sens que pour les applications utilisant les fichiers.

Le bilan de traitement indique en nombre et en montant les enregistrements traités en entrée et en sortie du programme. Ces indications peuvent servir à la fois de contrôle sur le bon fonctionnement du système, ou de valeur statistique pour des besoins particuliers.

LE DOSSIER DE PROGRAMME

EXPOSE DU PROBLEME

Il faut savoir évidemment de quoi il s'agit et comment le résoudre.

Un résumé succint, doit permettre de poser clairement le problème. Sa résolution technique n'en sera que plus aisée.

MASQUES D'ECRAN

Dessiner sur grille d'écran les données telles qu'elles seront produites depuis le programme.

Cette étape est très importante, car elle oblige son auteur à simuler le travail tel qu'il devra s'effectuer.

Quand on est d'accord avec l'ensemble des plans d'impression à obtenir, on a déjà franchi la moitié du chemin.

Il y a 3 parties à traiter :

- les dialogues conversationnels - questions/réponses ;
- les résultats (issus des fichiers, des tableaux DATA, calcul, etc.) ;

- les messages d'erreurs programmes.

Chaque partie est un sous-ensemble pouvant être séparément résolu. Ces trois parties profilent le découpage des séquences de programmation à écrire plus tard.

L'ORGANIGRAMME

Il permet de matérialiser par bloc de fonction, toutes les tâches que le programmeur doit prendre en charge.

Il donne une vision chronologique des événements à traiter, c'est pratiquement la radioscopie du futur programme. L'organigramme est indispensable pour donner, à priori, une cohérence certaine entre les séquences de programmation.

IDENTIFICATION DES VARIABLES

Un programme nécessite bien souvent, surtout en gestion, l'utilisation d'un nombre important de zone d'information, appelé plus couramment dans le monde des P.S.I. : variables.

Chaque variable doit être identifiée par un nom de baptême.

Au fur et à mesure de l'avancement d'un problème, il y a rapidement inflation du stock de variables disponibles, aussi pour y mettre de l'ordre, leur inventaire systématique est une sage précaution.

La fiche d'identification présentée plus loin comporte les paramètres nécessaires à la personnalisation de chaque variable.

L'utilisation récursive d'une variable nécessite dans bien des cas, l'initialisation de son contenu. Pour celà deux fonctions importantes de la programmation doivent être assumées avec clairvoyance :

- la mise à l'état initial de son contenu (M.E.I.) ;
- la remise à l'état initial de son contenu (R.E.I.).

On peut situer assez facilement le niveau d'exécution d'une M.E.I., par contre, il est beaucoup plus difficile de localiser la R.E.I., et c'est encore moins évident de déterminer parmi le nombre de variables utilisées, celles qui doivent faire l'objet d'une M.E.I. et de repérer parmi celles-ci, celles pour qui une R.E.I. est obligatoire.

Un tableau synoptique des variables présenté plus loin, donne les fonctions M.E.I. et R.E.I. à effectuer ou pas, selon le type de la variable.

FICHE D'IDENTIFICATION DES VARIABLES

NOM SYMBOLIQUE

L'étiquette contient en elle-même, le caractère de la variable,

A numérique (entier + décimale

A% numérique (entier seulement)

A$ alpha numérique (chaîne de caractères)

TYPE DE LA VARIABLE

Aiguillage : valeur relative exprimant au moins 2 états possibles

Compteur : incrémenté par les opérateurs arithémtiques ou fonctions mathématiques

Constante : valeur mise au début de programme à un état initial donné, restant inchangée pendant tout le déroulement du programme

Index : contient l'adresse relative d'une donnée définie dans un tableau ou une DATA

Indice : incrémenté par une fonction FOR....NEXT

Tableau : initialisé par DIM et/ou READ...DATA

Variable : valeur indéterminée, utilisable à différents niveaux du programme pour des conditions de traitement différentes.

ORIGINE

Clavier, programme, cassette, disquette.

LONGUEUR

Préciser entiers et décimales 2+2

entier 2

OBSERVATIONS

Préciser le rôle de la variable, ses différentes valeurs contenues.

DATE	NOM DU PROGRAMME :	PAGE /

IDENTIFICATION des VARIABLES

NOM SYMBOLIQUE	TYPE (*)	ORIGINE Clavier	ORIGINE Programme	ORIGINE Cassette	ORIGINE Disquette	LONG.	OBSERVATIONS

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable.

SYNOPTIQUE DES VARIABLES AVEC LES FONCTIONS MEI-REI

TYPE DE VARIABLE	M.E.I.	R.E.I.
AIGUILLAGE	Oui	Oui
COMPTEUR	Facultative	Facultative
CONSTANTE	Oui	Non
INDEX	Oui	Facultative
INDICE	Oui	Facultative
TABLEAU	Oui	Oui (RESTORE)
VARIABLE	Facultative	Facultative

FEUILLES DE PROGRAMMATION

Elles ont un triple usage :

- Etablir les masques d'écran ;
- Supporter les lignes de programmation ;
- Donner en abscisse et en ordonnée, les valeurs décimales de chaque point de l'écran.

DATE | NOM DU PROGRAMME | PAGE /

FEUILLE DE PROGRAMMATION BASIC

LIGNE ECRAN

Numéro de ligne

NUMEROS DES COLONNES ECRAN

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

6/1979

ANNEXES

- Contrôle de saisie ;
- Contrôle de validité ;
- Contrôle de vraisemblance ;
- Contrôle de compatibilité interne (comparaison de variables internes au programme) ;
- Contrôle de compatibilité externe (comparaison de variables internes avec des variables issues de fichiers entrée).

- Liste des messages d'erreur ;
- Développer la liste des différents messages d'erreur. Editer les messages d'erreur à partir de la même séquence programme si possible.

- Aspect maintenance. Liste des variables avec leur niveau programme d'utilisation pour celles susceptibles d'avoir leur contenu modifié d'un traitement à l'autre.

EXEMPLE D'UTILISATION

EXPOSE DU PROBLEME

DE QUOI S'AGIT-IL ?

De comptabiliser certaines fournitures de skis (skis, fixations, chaussures). Les commandes sont effectuées par différents magasins spécialisés dans la vente de ce type de matériel. Chaque magasin représente un client.

Pour chaque client qui a effectué une commande, on calcule et on édite une facture sur l'écran.

Périodicité des traitements	= à la demande
Volume des transactions	= minimum 1 facture maximum 10 factures par session
Délai de remise des résultats	= 24 heures après le traitement, envoi des factures par courrier.

Ouvrons là, une parenthèse pour indiquer qu'à partir d'un exemple simple, notre but est de révéler l'intérêt de la méthode.

Avant de poursuivre plus avant l'étude, on peut s'accorder un temps de réflexion et se demander si le matériel à notre disposition peut remplir les conditions demandées.

A priori, un seul P.S.I. de taille modeste peut supporter le volume des transactions à traiter, et répondre au délai exigé, fermons la parenthèse.

Le problème choisi est volontairement simplifié afin d'en faciliter la démonstration.

Ce guide étant un outil de réflexion, il peut et doit être un tremplin permettant la réalisation de dossiers techniques sur des applications informatiques infiniment plus complexes.

La qualité du contenu de ces dossiers dépendra de la part de chacun, des facultés d'adaptation à utiliser un tel outil et, bien entendu, du savoir faire professionnel acquis par l'expérience.

COMMENT TRAITER LE PROBLEME ?

Le fichier "client" fixé à 10 magasins est stocké en mémoire vive.

Le programme demande en début de session, la date de facturation commune à l'ensemble des factures, et le dernier numéro de facture imputé (relatif au traitement précédent) ; la toute première fois, il faut introduire zéro.

Le programme se charge ensuite de l'incrément par +1 pour chaque nouvelle facture.

A chaque nouvelle facture à éditer, les informations suivantes doivent être introduites préalablement au clavier.

Le numéro de code client

Il permet de rechercher dans le fichier "clients" les données d'identification du magasin concerné valeur comprise de 1 à 10.

La quantité des articles commandés

Introduire Ø si quantité nulle. Dans ce cas le programme n'éditera pas la ligne correspondante.

Une confirmation des données introduites est demandée par le programme.

Ensuite calcul et édition de la facture. Une routine de formatage des montants avant leur édition est prévue.

Cette routine est précieuse pour les P.S.I. non pourvus du PRINT-USING.

A chaque fin d'édition d'une facture, le programme demande si c'est la dernière.

Si la réponse est non, le programme retourne sur la séquence d'introduction d'un nouveau code client.

Si la réponse est oui, le programme affiche en contraste inverse

"PROGRAMME TERMINE"
READY

REMARQUES

Pour éviter de présenter un programme trop complexe et trop chargé en lignes d'écriture, nous avons délibérément écarté certaines fonctions telles que :

- contrôle des données introduites au clavier ;
- routine d'effacement partiel de l'écran ;
- routine d'édition de messages d'erreur ;
- écriture de la facture sur imprimante ;
- écriture du dernier numéro de facture imputé sur fichier.

L'ORGANIGRAMME GENERAL DU PROGRAMME

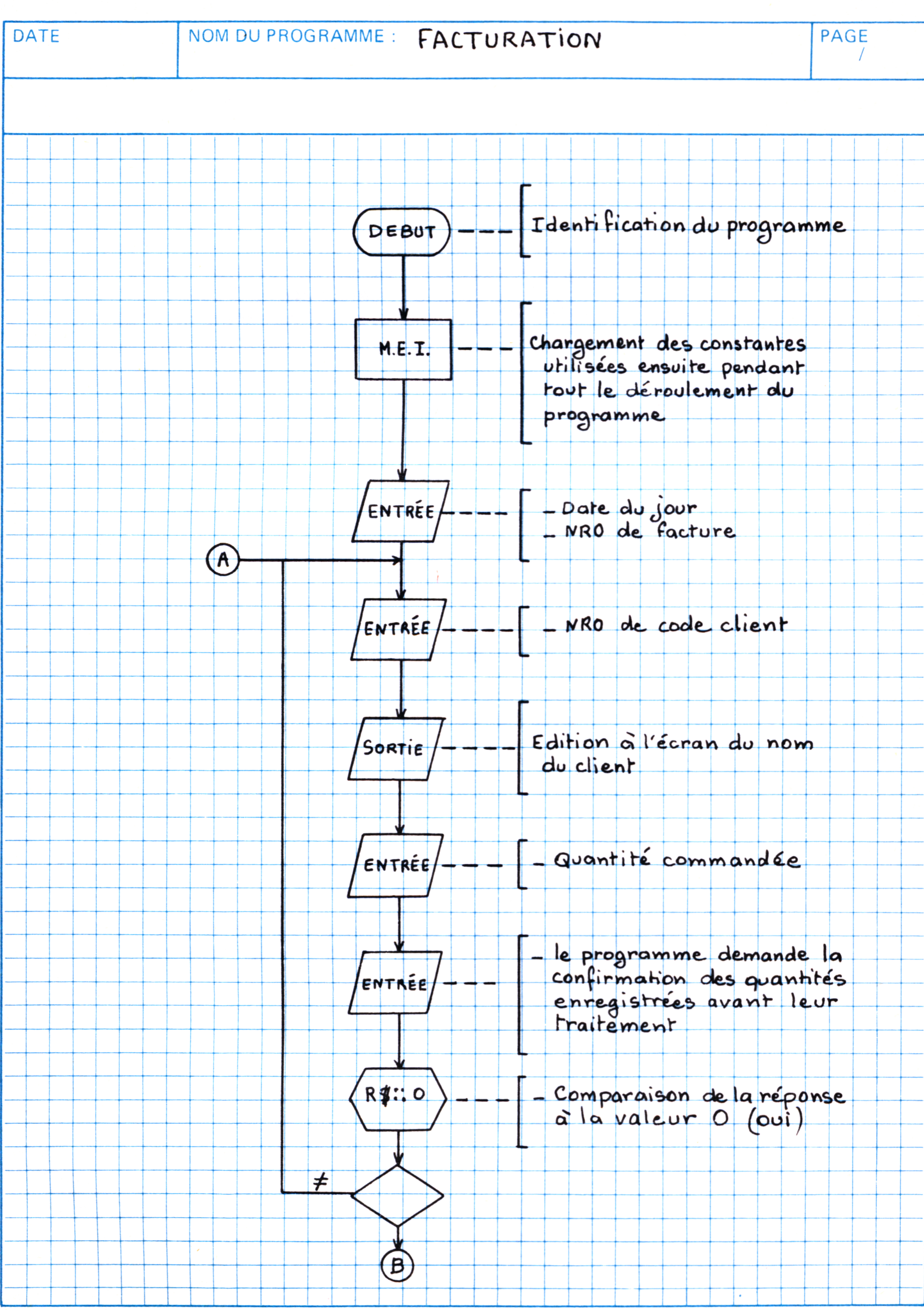
Avant d'écrire une ligne de programme il est recommandé de passer d'abord par l'élaboration des structures générales. Tout comme un architecte dessine le plan d'ensemble avant de réaliser les plans d'exécution.

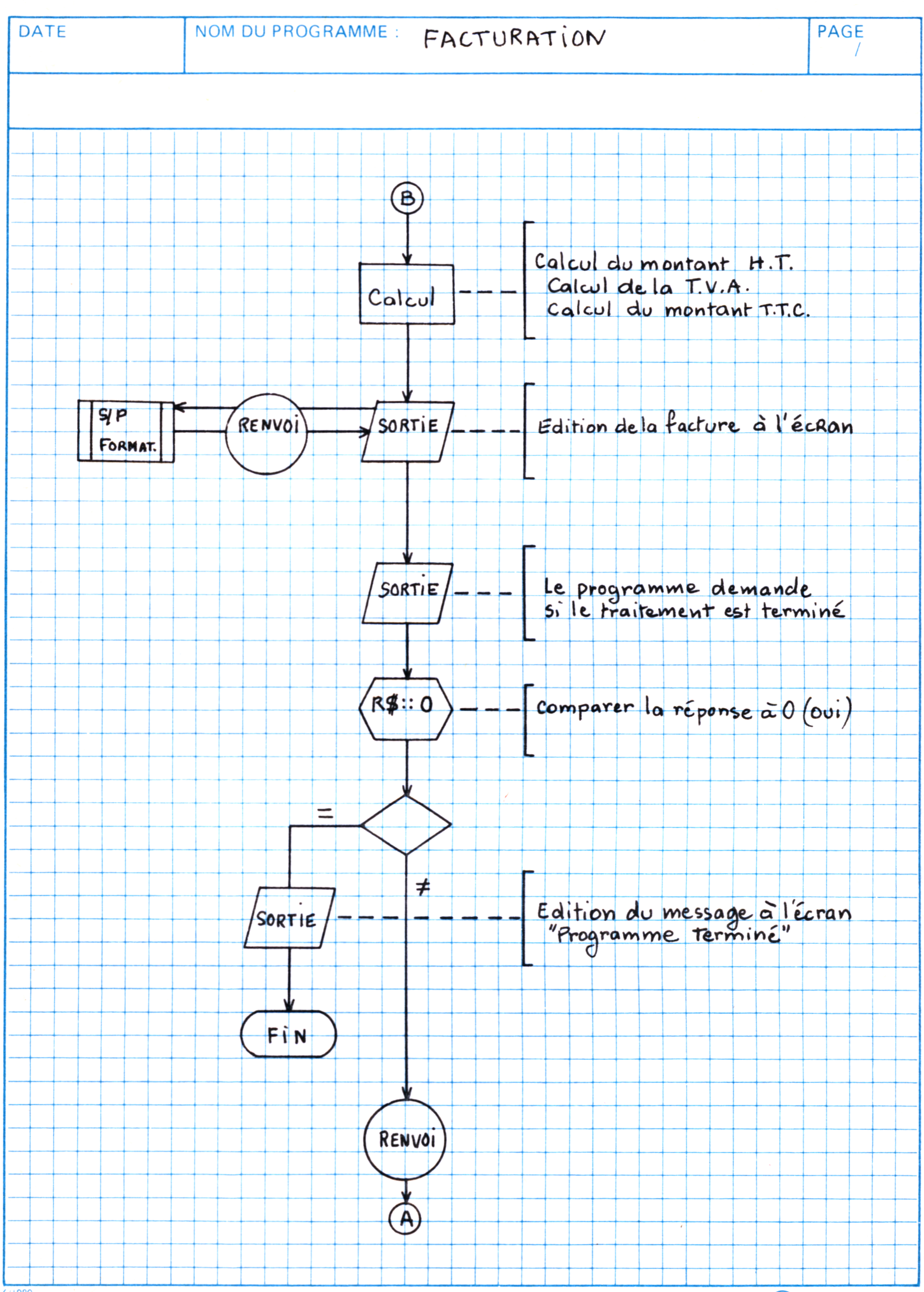
L'organigramme permet de faire l'inventaire de toutes les fonctions nécessaires, il révèle aussi celles qui peuvent manquer

lorsqu'on prend soin de parcourir tous les cas de figure possibles.

L'exemple présenté est plutôt linéaire donc assez simple à comprendre ; les renvois du genre A , B , sont utiles lorsque l'organigramme s'étend sur plusieurs pages.

DATE | NOM DU PROGRAMME : FACTURATION | PAGE 1




DATE
NOM DU PROGRAMME : FACTURATION
PAGE
/
B
Calcul
Calcul du montant H.T.
Calcul de la T.V.A.
Calcul du montant T.T.C.
S/P
FORMAT.
RENVOI
SORTIE
Edition de la facture à l'écran
SORTIE
Le programme demande
si le traitement est terminé
R$:: 0
comparer la réponse à 0 (oui)
=
≠
SORTIE
Edition du message à l'écran
"Programme terminé"
FIN
RENVOI
A

LES GRILLES D'ECRAN

Les grilles d'écran sont des photographies de l'information traitée. Il est indispensable de les dessiner avant d'entreprendre leur réalisation programmée.

Lorsque tous les plans sont établis il suffit de les ordonnancer. A ce niveau la programmation devient "péripétie". Le scénario étant bâti, le film peut se dérouler. Silence on tourne ! Moteur.

Le scénario de la facturation comporte deux plans (pages suivantes). Le premier représente l'introduction des données indispensables aux calculs d'une facture. Avant de passer au plan suivant le programme demande la confirmation des valeurs introduites. Si une erreur s'est glissée, le programme retourne en début d'introduction.

Le deuxième plan représente une facture complète pour un client.

Le numéro de client permet de rechercher et d'éditer la raison sociale automatiquement grâce à un fichier indicé stocké dans le programme en zone DATA.

MASQUE 1

GRILLE D'ECRAN

8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

3276	FACTURATION
3280	
3284	-DATE DU JOUR.FORMAT JJMMAA : 210580
3288	-NUMERO DE FACTURE : 4
3292	
3296	-NUMERO DE CODE CLIENT : 1
3300	NOM-JEAN PAUL DUCHEMIN
3304	
3308	*** COMMANDES ***
3312	
3316	-ARTICLE - SKIS - QTE : 10
3320	
3324	-ARTICLE - FIXATIONS - QTE : 15
3328	
3332	-ARTICLE - CHAUSSURES - QTE : 30
3336	
3340	
3344	
3348	ETES-VOUS D'ACCORD SUR LES DONNEES
3352	INTRODUITES ?
3356	
3360	REPONDEZ O POUR OUI N POUR NON ?
3364	
3368	
3372	

8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

MASQUE 2

GRILLE D'ECRAN

8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

3276	CLIENT NRO. 1 JEAN PAUL DUCHEMIN
3280	
3284	
3288	
3292	FACTURE NRO. 004 DU 21/05/80
3296	
3300	QTE P.U. DOIT
3304	
3308	
3312	
3316	-SKIS 10 1200.00 12000.00
3320	
3324	-FIXATIONS 15 500.00 7500.00
3328	
3332	-CHAUSSURES 30 600.00 18000.00
3336	
3340	TVA 17.60 6600.00
3344	
3348	
3352	TOTAL A REGLER 44100.00
3356	
3360	DERNIERE FACTURE (O/N) ? O
3364	PROGRAMME TERMINE
3368	
3372	

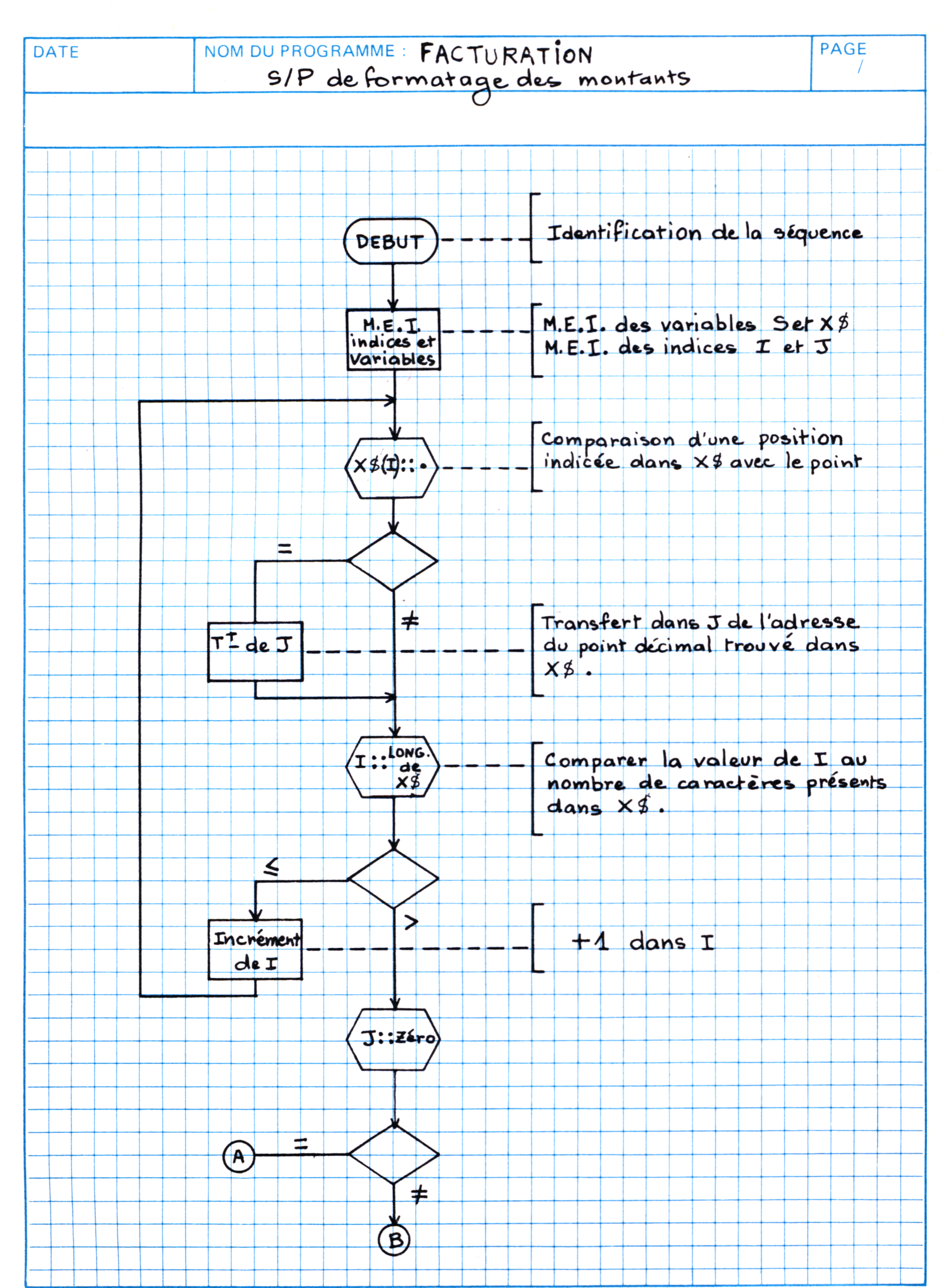
8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7

ORGANIGRAMMES DETAILLES

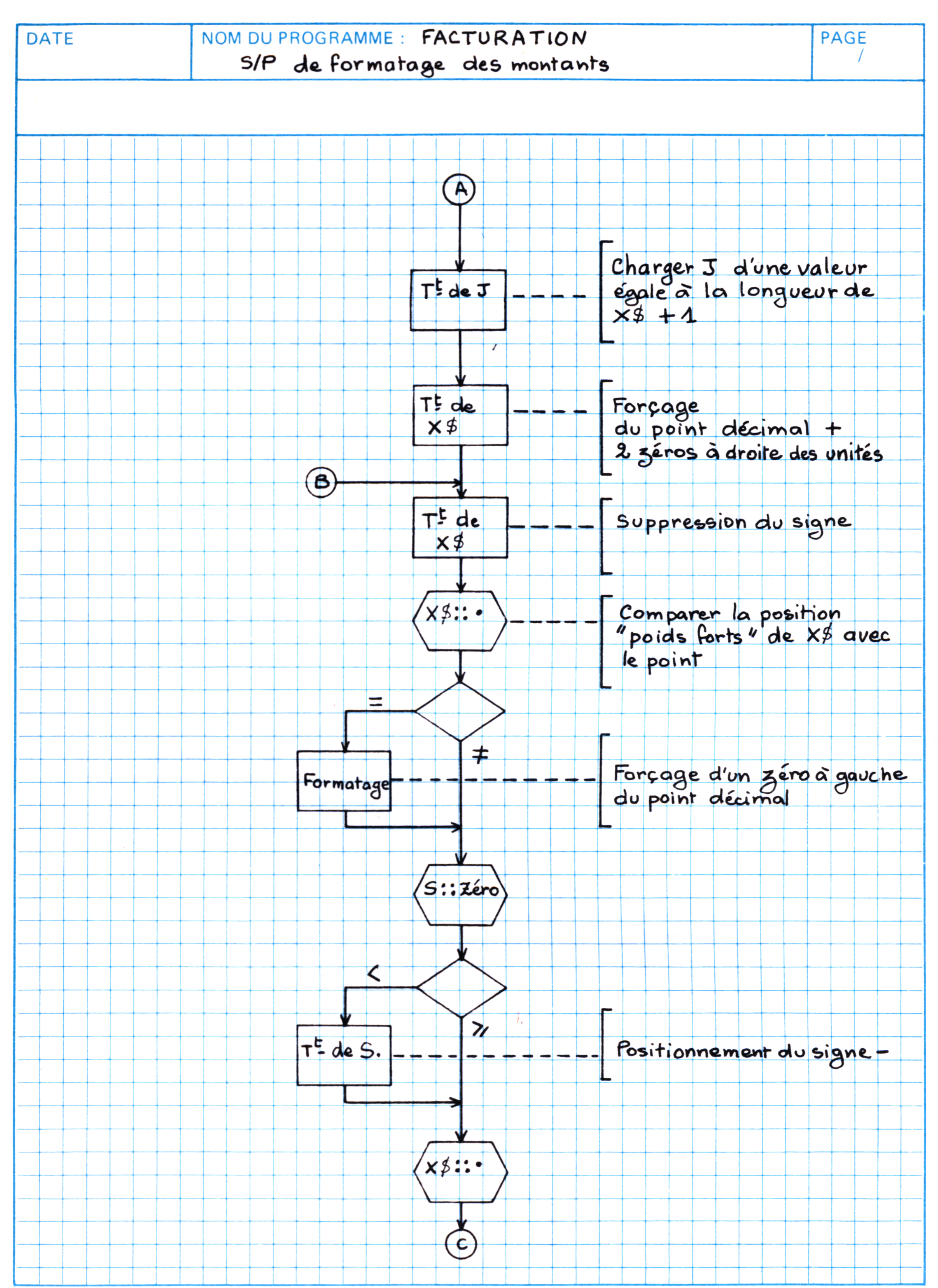
Il s'agit, fonction par fonction, d'établir un organigramme aussi précis que le nécessite la complexité du problème à traiter.

La routine de formatage des montants est une séquence complexe qui a toujours intérêt à être radiographiée. L'organigramme est, répétons-le un excellent outil en la matière pour inspecter et démonter les mécanismes les plus subtils.

DATE | NOM DU PROGRAMME : FACTURATION
S/P de formatage des montants | PAGE /



DATE | NOM DU PROGRAMME : FACTURATION
S/P de formatage des montants | PAGE /



DATE

NOM DU PROGRAMME : FACTURATION

S/P de formatage des montants

PAGE /

LA PROGRAMMATION

La tentation est forte de programmer directement sur P.S.I. Mais la rédaction préalable du programme et l'utilisation d'un support structuré et adapté au matériel choisi améliorent considérablement le travail du programmeur.

De sacrifier une heure de son temps à écrire les séquences sur un tel support peut en faire gagner cinq fois plus à la mise au point. (c'est un minimum). De plus, pour les utilisateurs de P.S.I. qui n'ont pas d'imprimante, la rédaction préalable des programmes n'est pas seulement conseillée, mais indispensable dès que l'on aborde des programmes de plusieurs dizaines de lignes.

DATE	NOM DU PROGRAMME FACTURATION	PAGE 6 /

FEUILLE DE PROGRAMMATION BASIC

Numéro de ligne — NUMEROS DES COLONNES ECRAN

```
1100 REM * ROUTINE DE FORMATAGE
1110 REM * DES MØNTANTS A EDITER
1120 J=Ø:REM * M.E.I. INDICE J
1130 S=INT(S*1ØØ+.5)/1ØØ
1140 X$=STR$(S)
1150 FOR I=1TOLEN(X$)
1160 IF MID$(X$,I,1)="."THEN J=I
1170 NEXT I
1180 IF J=Ø THEN J=LEN(Y$)+1:X$=X$+".ØØ"
1190 X$=MID$(X$,2,J+2):REM * RECADRAGE
1200 REM * DE X$ EN SUPPR.LE SIGNE
1210 IF MID$(X$,1,1)="."THEN X$=RIGHT$(M1$,1
)+X$

1230 IF S<Ø THEN X ="-"+X
1240 IF MID$ (X$) LEN (X$)-1,1)="."THENX$=X$+"Ø
"
1250 LS=LEN(X$)
1260 U=39-LS


1290 RETURN
```

6/1979

IDENTIFICATION DES VARIABLES

Il s'agit, en quelque sorte, de la nomenclature des variables utilisées dans un programme donné.

Pour le programme de facturation, 27 variables ont été recensées.

Dresser l'inventaire des zones de travail d'un programme est salutaire à plus d'un titre : d'une part la mise au point est plus rapide, d'autre part les modifications de programme proches ou lointaines de la réalisation initiale sont grandement facilitées par le rappel du rôle tenu de chaque variable.

Les critères d'identification retenus sont suffisamment explicites pour considérer ce support comme un auxiliaire précieux de la gestion d'un programme.

DATE	NOM DU PROGRAMME : FACTURATION	PAGE 1

IDENTIFICATION des VARIABLES

NOM SYMBOLIQUE	TYPE (*)	ORIGINE Clavier	Programme	Cassette	Disquette	LONG.	OBSERVATIONS
D$	variable	X				6	Date du jour pour l'ensemble des factures.
F1		X				≤ 3	NRo-de facture premier d'une série.
F2			X			≤ 3	NRo-de facture calculé
C1		X				≤ 2	NRo-de Code client
Q1		X				≤ 3	Nombre d'articles de skis
Q2		X				≤ 3	de fixations
Q3		X				≤ 3	de chaussures
P1	Constante		X			≤ 7	P.U. de l'article skis
P2			X			≤ 7	Fixations
P3			X			≤ 7	chaussures
D1	variable		X			≤ 8	total Q1 × P1
D2			X			≤ 8	Q2 × P2
D3			X			≤ 8	Q3 × P3
T1	Constante		X			≤ 4	T.V.A. (taux)
T2	variable		X			≤ 7	Montant de la T.V.A.
T3			X			≤ 8	Montant total de la facture
R$		X				1	Réponse OUI ou NON
N$			X			≤ 30	Nom du client
L$	Tableau		X				Fichier des clients en DATA
I	Indice		X			≤ 2	Boucle FOR/NEXT
S	variable		X			≤ 8	Zone de manoeuvre pour routine de formatage
X$	variable		X			≤ 8	Zone de manoeuvre pour routine de formatage

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable.

DATE	NOM DU PROGRAMME :	PAGE /

IDENTIFICATION des VARIABLES

NOM SYMBOLIQUE	TYPE (*)	ORIGINE Clavier	ORIGINE Programme	ORIGINE Cassette	ORIGINE Disquette	LONG.	OBSERVATIONS
L S	variable		X			1	contient la longueur de X$
U	〃		X			≤2	nombre de caractères à tabuler
J	〃		X			1	Zone de manœuvre pour la routine de formatage
Y $	〃		X			≤8	Zone de manœuvre pour l'édition de Q1,Q2,Q3
Y	〃		X			1	Nombre de caractères à tabuler

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable.

PRESENTATION DU LISTING DE PROGRAMME

La liste complète du programme de facturation présenté aux pages suivantes comporte des remarques insérées entre chaque fonction principale grâce à l'instruction Basic - REM - ; celle-ci permet une mise en page de la liste par paragraphe ; il représente chacun un sous-ensemble logique de fonction donnant plus de clarté au problème traité.

Sa relecture en est donc plus aisée surtout lors des phases successives de la mise au point. Les maintenances ultérieures (si besoin est) seront également facilitées par une localisation instantanée des niveaux de modification demandés.

Le découpage des séquences de traitement du programme se présente donc ainsi :

Lignes du programme	Commentaires
5 à 80	Identification du programme
100 à 120	Mise à l'état initial des constantes
200 à 260	Introduction de la date du jour et du premier NRO de facture
270	Mémorisation du premier NRO de facture
300 à 330	Introduction du NRO de code client
340 à 380	Lecture du fichier des noms et transfert du nom trouvé dans une zone de manoeuvre
390 à 400	Edition du nom
500 à 560	Introduction des commandes
600 à 680	Confirmation des données introduites - REI de LØ
700 à 770	Calcul de la facture
800 à 890	Edition de la facture
900 à 1010	Séquence de fin de travail - REI de LØ
1100 à 1290	Sous-programme de formatage des montants à éditer
2000 à 2030	Implantation du fichier des noms de client.

```
10 REM *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
15 REM *                              *
20 REM *    EXEMPLE D'UTILISATION     *
30 REM *       DU GUIDE PRATIQUE      *
40 REM * "REALISATION DES PROGRAMMES"*
45 REM *                              *
50 REM *    FACTURATION SIMPLIFIEE    *
60 REM *  AUTEUR : MICHEL BENELFOUL   *
65 REM *                              *
70 REM * COPYRIGHT EDITIONS DU P.S.I.*
80 REM *          ET L'AUTEUR         *
85 REM *                              *
90 REM *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
99 :
100 REM **** M.E.I. DES CONSTANTES****
101 REM ******************************
110 P1=1200.00:P2=500.00:P3=600.00
120 T1=17.60:M1$="10"
199 :-------------------------------
200 REM **** INTRODUCTION DATE ET*****
201 REM ****PREMIER NRO DE FACTURE****
202 REM ******************************
220 PRINT;TAB(10);"FACTURATION":PRINT
230 PRINT"-DATE DU JOUR-FORMAT JJMMAA- ";
240 INPUT D$
250 PRINT"-NUMERO DE FACTURE";SPC(11);
260 INPUT F1
270 F2=F1 :REM###INITIAL. COMPT. NRO. FACT.###
299 :-------------------------------
300 REM **** INTRODUCTION DU NRO DE***
301 REM *********CODE CLIENT*********
302 REM ******************************
310 PRINT
320 PRINT"-NUMERO DE CODE CLIENT       ";
330 INPUT C1
399 :-------------------------------
400 REM ******* RECHERCHE DU NOM******
401 REM *******DANS UNE ZONE DATA*****
402 REM ******************************
410 FOR I=1 TO 10
420 :   READ L$(I)
430 :   IF I=C1 THEN N$=L$(I)
440 NEXT I
450 PRINT:PRINT" CLIENT- ";N$:PRINT
499 :-------------------------------
500 REM **** INTRO.DES COMMANDES******
501 REM ******************************
505 PRINT"+++ COMMANDES +++":PRINT
510 PRINT"-ARTICLES   SKIS         QTE  ";
520 INPUT Q1:PRINT
530 PRINT"-ARTICLES   FIXATIONS    QTE  ";
540 INPUT Q2:PRINT
550 PRINT"-ARTICLES   CHAUSSURES   QTE  ";
560 INPUT Q3:PRINT:PRINT:PRINT
599 :-----------------------------------
```

```
600 REM **** LE PROGRAMME DEMANDE*****
601 REM *******LA CONFIRMATION********
602 REM ***DES DONNEES INTRODUITES****
603 REM ******************************
610 PRINT"ETES-VOUS D'ACCORD SUR LES DONNEES"
620 PRINT"INTRODUITES ?":PRINT
630 PRINT"REPONDEZ O POUR OUI N POUR NON ";
640 INPUT R$
650 IF R$="O"THEN 700
660 RESTORE:GOTO 300:REM###RETOUR SUR SEQ.D'INTRO###
699 :————————————————————
700 REM **** CALCUL DE LA FACTURE*****
701 REM ******************************
710 D1=P1*Q1:D2=P2*Q2:D3=P3*Q3
720 T=D1+D2+D3:REM###TOTAL H.T.###
730 T2=T1*T/100:REM###TVA###
740 T3=T+T2:REM###TOTAL A PAYER###
750 F2=F2+1:REM###NRO FACTURE +1###
799 :————————————————————
800 REM **** EDITION DE LA FACTURE****
801 REM ******************************
810 PRINT"CLIENT NRO.";C1;" ";N$:PRINT:PRINT
820 PRINT"FACTURE NRO. ";F2;"  DU ";:PRINT LEFT$(D$,2);"/";
825 PRINT MID$(D$,3,2);"/";
830 PRINT RIGHT$(D$,2):PRINT
840 PRINT TAB(15);"QTE";SPC(4);"P.U.";SPC(6);"DOIT":PRINT:PRINT
845 S=D1:GOSUB 1100:Y$=STR$(Q1):Y=LEN(Y$):Y=18-Y
847 IF Q1<=0 THEN PRINT:GOTO 855
850 PRINT"-SKIS";TAB(Y)Q1;P1;TAB(U)X$:PRINT
855 S=D2:GOSUB 1100:Y$=STR$(Q2):Y=LEN(Y$):Y=18-Y
857 IF Q2<=0 THEN PRINT:GOTO 865
860 PRINT"-FIXATIONS";TAB(Y)Q2;P2;TAB(U)X$:PRINT
865 S=D3:GOSUB 1100:Y$=STR$(Q3):Y=LEN(Y$):Y=18-Y
867 IF Q3<=0 THEN PRINT:GOTO 875
870 PRINT"-CHAUSSURES";TAB(Y)Q3;P3;TAB(U)X$:PRINT
875 S=T2:GOSUB 1100
880 PRINT TAB(16);"TVA  ";T1;TAB(U)X$
885 S=T3:GOSUB 1100:PRINT
890 PRINT TAB(13);"TOTAL A REGLER";TAB(U)X$:PRINT
899 :————————————————————
900 REM **** TEST DERNIERE FACTURE****
902 REM ******************************
910 INPUT"DERNIERE FACTURE (O/N) ";R$
920 IF R$="O" THEN 1000
930 RESTORE:GOTO 300
999 :————————————————————
1000 REM******************************
1001 REM*****FIN DU PROGRAMME*********
1002 REM******************************
1010 PRINT"PROGRAMME TERMINE"
1020 END
1099 :————————————————————
```

.../...

```
1100 REM **** ROUTINE DE FORMATAGE****
1101 REM ****DES MONTANTS A EDITER ***
1102 REM ******************************
1120 J=0:REM###M.E.I.INDICE J###
1130 S=INT(S*100+.5)/100
1140 X$=STR$(S)
1150 FOR I=1 TO LEN(X$)
1160 :   IF MID$(X$,I,1)="." THEN J=I
1170 NEXT I
1180 IF J=0 THEN J=LEN(X$)+1:X$=X$+".00"
1190 X$=MID$(X$,2,J+2):REM###RECADRAGE
1200 REM:DE X EN SUPPRIMANT LE SIGNE###
1210 IF MID$(X$,1,1)="." THEN X$=RIGHT$(M1$,1)+X$
1230 IF S<0 THEN X$="-"+X$
1240 IF MID$(X$,LEN(X$)-1,1)="." THEN X$=X$+"0"
1250 LS=LEN(X$)
1260 U=39-LS
1290 RETURN
1299 :----------------------------------
2000 REM ******************************
2001 REM *******FICHIER CLIENTS*******
2002 REM ******************************
2009 :
2010 REM ###LES NROS DE CODE CLIENT###
2020 REM ###SONT LES 2 DERNIERS ######
2030 REM ####CHIFFRES DU NRO LIGNE####
2040 REM ###EXEMPLE:CLIENT 8=2108#####
2099 :
2101 DATA JEAN PAUL DUCHEMIN
2102 DATA ARTHUR PERE & FILS
2103 DATA ETABLISSEMENT RIQUET
2104 DATA MAUDUIT SHOP
2105 DATA J.C.KILLY
2106 DATA PERILLAT CENTER
2107 DATA MERIBEL BABIOLES
2108 DATA COURCHEVEL VOG
2109 DATA SERRE-CHEVALIER IN
2110 DATA VAL D'ISERE COMPETITION
```

MISE AU POINT
ET MAINTENANCE

LES MODALITES DE LA MISE AU POINT D'UN PROGRAMME

Lorsque le dossier de programmation est constitué, l'introduction des séquences de traitement dans l'ordinateur peut commencer.

La procédure d'introduction doit se faire par étape. Chaque étape est représentative d'une ou de plusieurs séquences de traitement liées entre-elles par une relation logique de fonctionnement.

Exemple : Dans notre programme de facturation, le test d'édition du nom de client par rapport à son code introduit au clavier, nécessite la présence dans la mémoire des séquences suivantes :

- séquence 300 à 330 — introduction du code client
- séquence 340 à 380 — lecture du fichier des noms de client
- séquence 390 à 400 — édition du nom
- séquence 2000 à 2030 — implantation du fichier des noms de client.

Chaque fin de test doit être ponctué par la sauvegarde du programme sur cassette ou disquette avant de passer à l'étape suivante.

Ceci est une sage précaution nous évitant de perdre le bénéfice du travail réalisé à cause d'une micro coupure intempestive pouvant détruire les données en mémoire.

Pour les séquences de traitement dont le test complet nécessite plusieurs sessions d'introduction des données, la construction d'un jeu d'essais simulant toutes les conditions réelles du travail doit être envisagée.

Par exemple, dans le contexte de notre programme de facturation, le scénario de test d'édition de la facture peut s'appuyer sur une table de décision limitée couvrant toutes les hypothèses relatives à l'introduction des commandes.

Si vous possédez une imprimante, la constitution d'un référentiel d'essais est conseillé. Les sorties illustrées des masques d'écran sont la garantie du bon fonctionnement du programme ; lors des maintenances ultérieures, elles servent d'interface logique entre l'ancienne et la nouvelle version du programme.

Construisons la table avec les éléments suivants :

1- **Enoncé des conditions.** 3 éléments constitués de paires de skis, de fixations, de paires de chaussures.

2- **Les règles.** 3 conditions = 2^3 combinaisons, ce qui fait 8 règles possibles de quantité à introduire.

3- **Enoncé des actions.** 8 sessions d'introduction des commandes relatives aux 8 règles de traitement.

		Les règles							
		0	1	2	3	4	5	6	7
Enoncé des conditions	Ski	non	oui	oui	oui	non	non	non	oui
	Fixations	non	non	oui	non	oui	oui	non	oui
	Chaussures	non	non	non	oui	non	oui	oui	oui
Enoncé ses actions	Session 1	X							
	Session 2		X						
	Session 3			X					
	Session 4				X				
	Session 5					X			
	Session 6						X		
	Session 7							X	
	Session 8								X

LA MAINTENANCE DU PROGRAMME

La maintenance du programme est un événement fréquent et banal, inhérent à toutes les applications traitant des données, dont les règles et les valeurs sont tributaires de phénomènes extérieurs à l'application.

Par exemple, la valeur du taux de la T.V.A. est sujet à modification. Le principe même de son application peut être remis en cause.

Aussi la constitution d'un support d'information adapté aux règles de maintenance doit permettre le recensement anticipé des données et des fonctions du programme susceptibles d'être modifiées.

Le dossier de maintenance doit comporter :

- le dossier de programme ;
- la liste des fonctions et des variables concernées par la maintenance ;
- les niveaux d'intervention dans le programme en précisant les paragraphes et les numéros de ligne ;
- les modifications réellement effectuées ;
- un référentiel d'essais obtenu selons les modalités de mise au point déjà évoquées ;
- mise à jour du numéro de version et la date du nouveau programme avant son écriture sur cassette ou disquette ;
- rééditer une liste complète du programme ;
- archiver la liste et le référentiel d'essais de la version précédente.

Achevé d'imprimer en janvier 1981
sur les presses de l'imprimerie Laballery et Cie
58500 Clamecy
Dépôt légal : 1er trimestre 1981

N° d'impression : 19904
N° d'édition : 86470-17-4
ISBN : 2-86470-017-4

GUIDE PRATIQUE

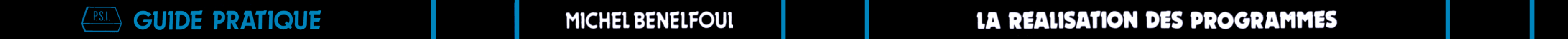
LA RÉALISATION DES PROGRAMMES

Ce guide pratique est destiné aux utilisateurs de Petits Systèmes Individuels qui, après avoir appris le Basic, sentent le besoin d'une approche méthodique de la réalisation des programmes : définition du problème, étude de la solution, programmation, mise au point, maintenance. Un exemple complet, une facturation simple, illustre les différentes étapes proposées.

Editions du P.S.I.
Boîte Postale 86
F - 77400 Lagny/Marne

Graphisme : Françoise Guillot

ISBN 2-86470-017-4 — Imprimé en France


P.S.I.
GUIDE PRATIQUE
MICHEL BENELFOUL
LA REALISATION DES PROGRAMMES



Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE

https://acpc.me/